Yelp Review Analysis

Shubham Kokane(ssk203), Yuehan Zhang(yz712)

Abstract – **Our goal in this project is to build an application that can provide a recommendation of a restaurant to the user based on the location and cuisine that he is interested in. We are using Yelp's Dataset for this project which is more than 5 GB. We plan to use CNN and LSTM models for this project which is a famous approach used by google and Spotify's recommendation systems.**

I. INTRODUCTION

Yelp is currently the most widely used restaurant and merchant information software across the United States. However, Yelp only provides us with a holistic view of restaurants, such as giving overall review scores or ratings and only a few reviews out of thousands of reviews. To improve Yelp users' experience, we dived deep into Yelp's open datasets and other data centers to retrieve useful information. Utilizing our complementary plug-in prototype, yelp users, whether they are business owners or consumers, can find information that better meets their preferences. Specifically, our group mainly focused on improving the customers' understanding of merchants, market knowledge for new business owners, and existing merchants' awareness of restaurants' features.

II. PROJECT DESCRIPTION

A. Vision:

In the first part of this work, we are examining the fake review detection data, which consists of 350,000 user reviews. The true and fake reviews in the data set help us train a model that predicts if a given review is fake or not. In the second part of this work, we examined Yelp's merchant review data in the hope of retrieving useful information for customers to understand a particular merchant better and for merchants to improve their businesses. We obtained our data from Yelp's online data challenge, utilized machine learning techniques and natural language processing tools to retrieve insights from the data and fit statistical models to the data so that we could access the most relevant keywords in the reviews that affect review scores. In the last part of our work, we have incorporated the household income dataset, which describes the income information in 6 states, namely Pennsylvania, Nevada, North Carolina, Illinois, Ohio, and Arizona. Combined with the merchant price range in our Yelp dataset, we mapped the average income and the average price range of restaurants in each county to help our new restaurant owners realize the relationship between the two, determine the potential size of the customer market, and ultimately to develop optimal pricing strategies.

B. About data:

This dataset includes information on reviews, users, businesses, checkin, photos, and tips. The dataset is 5.79 gigabytes uncompressed in *json* format (6 *json* files, including $business.json$, $check-in.json$, $photos.json$, $review.json$, $tip.json$ and $user.json$) Our project mainly focused on reviews, users, and business datasets from Yelp's open data source. For reviews, we kept the unique id of each review, the user id for the people writing the review, the business id for the restaurants that the user wrote it for, the review content, the rating according to the review, and the date when the review was written. We also added in one geolocation character into the review data, which indicates the restaurant's location where the review implied. We kept the user id, the number of reviews that a user has written, the time since a user joined Yelp, and the average ratings of reviews that a user has written from the $user.json$ file. Last by not least, we only included business id, name, business categories, geolocation information (city, state, postal codes, latitude, longitude), price range, ratings, number of reviews, and whether the restaurant is open or not from the business dataset.

C. Data Cleaning:

We only kept business information from 6 states in the United States. Since we are only interested in restaurants, we filtered out all other types of businesses in the business dataset. Moreover, we noticed that the closed restaurants did not contain much useful review information, so we removed restaurants that were not open by the 'is _open' column. With a cleaned business dataset, we matched business ids in both business and review datasets to remove all the reviews that were irrelevant to the restaurants we cleaned. As a result, we obtained cleaned versions of review and business datasets.

D. Steps:

- Data Observation
- Data Preprocessing
- Feature Engineering
- Training and Model Generation
- Model evaluation and application development

III. PROJECT DESIGN

A. SQL Database:

With cleaned reviews, users, business, income zip code, and fake review labeled data in five separated *csv* files, it is essential to connect them in order to reduce the time required for merging and extracting data for later models. It also enables scalability in the future if we have to merge or update more data into these datasets. With all the benefits, we built a SQL database with four out of five *csv* files (reviews, business, income zip code, and users) utilizing the following relationships:

- A business may contain multiple reviews.
- An user may write many reviews for different restaurants.
- Each review can only be written by one user about one restaurant. Review id is the primary key for the

review_fact_table, which connects the user's table and business table.

- Users table and review_fact_table has N: 1 relationship, meaning one user id has N review ids, but one review id can only define one user id.
- Business table and review_fact_table has an N:1 relationship, indicating one business id has N review ids, but one review id can only define one business id.
- Income zip code table and review_fact_table has an N:1 relationship since a zip code can have N review ids, but 1 review id can only imply one zip code.
- $Business_id$, $user_id$, and *zipcode* are all foreign keys in the review_fact_table.
- All *user_id*, *business_id*, and *review_id* are mixture of letters and numbers, so we used *VARCHAR* instead of integers. Ids are pre-populated by Yelp, so we do not need to create unique IDs ourselves.

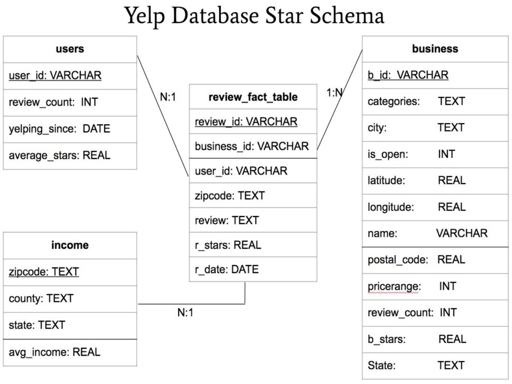


Fig. 1. Yelp Database Star Schema

B. Exploratory Statistics:

Before jumping right into our machine learning models, we explored and familiarized ourselves with these datasets through graphs and some preliminary analysis. For each of the above datasets, we tried to find patterns and potential problems which could be useful information or traps in the later machine-learning processes. Three main areas we explored are star ratings distribution, restaurant categories, and income versus restaurants' price range in each of the six states.

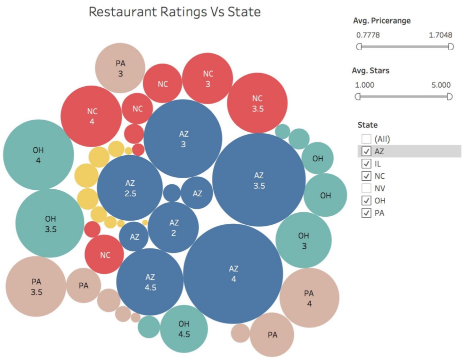


Fig. 2. Restaurant Rating Vs State

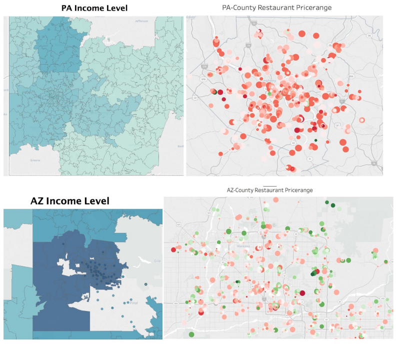

PA Income Level
PA-County Restaurant Pricerange
AZ Income Level
AZ-County Restaurant Pricerange

Fig. 3. Income level compare

IV. METHODOLOGY

A. Data Analysis:

By working with data analysis, we found that our data is unbalanced. The number of true reviews is about 10 times more than the number of fake reviews. To train a proper machine learning model, we solved this problem by replicating fake reviews ten times to have the same sample as the number of true reviews.

By graphing the distribution of the length of each review, we also noticed that the length of most reviews is within 500, which helped us to choose the best embedding parameters for our deep learning model.

Reviews are also written in other languages, such as Chinese and French. Since these comments are rare, and we are only interested in English, we removed all foreign reviews.

B. Models:

- Model 1: Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM)

Pre-process: Before the training, we transformed our text data into numerical form. We utilized Kera's pre-processing module called a tokenizer, which pre-processes text automatically by its insertion functions. The first step of the tokenizer is to split the text by space, filter out all punctuations, and convert the text to lowercase. Keras then transforms each word into numerical representations. In a given dataset, Keras presents the most common word as 1, the second most common word as 2, and so on. Since rare words cannot provide useful information for neural network models but only add noise, Keras is very useful in ignoring these rare words.

1) Embedding Layer: The layer expands each token to a larger vector, allowing the network to represent words meaningfully. We passed 20,000 as the first argument, which takes in the most important 20,000 terms in the model. This parameter is flexible for tuning. We chose 20,000 because there is a good tradeoff between computational expenses and accuracy. To obtain a significant improvement, we would need a much longer training time.

2) Convolutional Layer: The layer passes a filter over the data and calculates a higher-level representation. The convolutional layer has performed surprisingly well for text, too. It is also faster because the different filters can be calculated independently. By adding a convolutional layer

before the Long Short-Term Memory layer (LSTM), we allowed LSTM to learn sequences of chunks instead of sequences of words.

3) Dropout Layer: A dropout layer is applied before the convolutional layer because we want to reduce a certain amount of data before feeding it into the neural networks to avoid overfitting problems.

4) Max-pooling Layer: A max-pooling layer followed by convolutional neural networks is always efficient because it extracts a higher-level representation of the dataset, which indicates more information will be expressed by max pooling.

5) Long Short-Term Memory: LSTM is a one-layer recurrent neural network which is known to perform very well on text data. LSMT is designed to learn sequences of data since it learns terms where each word is related to those before and after it in a sentence. Therefore, it works well for our goals. In addition, LSMT allows the neural network to pay more attention to certain parts of a sequence and largely ignore words that are not useful.

- Model 2: Support Vector Machine Classifier

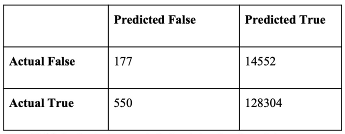
Pre-process: We used a feature extraction module called TfidfVectorizer, a scheme that transformed each review into a large sparse matrix with each cell representing a word and the frequency it appears in that review. Then TfidfVectorizer normalizes the counts by dividing the total number of times the word appears in all reviews. Setting ngrams to 2, we considered all phases that contain two words. By assumption, a larger ngram will perform better but requires a larger dataset. With a try and error, we determined a good tradeoff by using bigrams. We also set min_df to 3, which removes all the words that appear less than three times in the document. The reason is that words with a very rare appearance in the text are not useful for model improvement. Besides, it only adds noise to our model.

We trained two models: a deep learning model with convolution and long short-term memory layers using tokenized text data. Another is a linear support vector machine utilizing 2-gram vectorized text data. The cross-validation scores for deep learning and support vector machine models are 0.8976 and 0.8973. Though both methods have roughly the same performance, the support vector machine is much faster than the deep learning model. With a try and error, we applied ensemble methods to combine the results of the two techniques. As a result, the performance is improved significantly.

V. RESULTS

1) Fake Review Detection:

a) Support Vector Machine: We tried to train a support vector machine on the unbalanced dataset. As demonstrated from the confusion matrix, the model has a higher preference for true reviews.

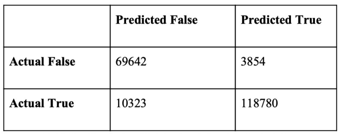


	Predicted False	**Predicted True**
Actual False	177	14552
Actual True	550	128304

Table1: Confusion Matrix before balancing the training set

Fig. 4

To train a proper machine learning model, we duplicated fake reviews ten times to have the same amount of samples as the number of true reviews. The result under the confusion matrix is listed below.

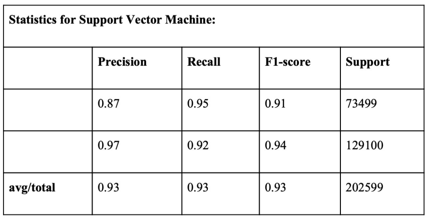


	Predicted False	**Predicted True**
Actual False	69642	3854
Actual True	10323	118780

Table2 : Confusion Matrix after balancing the training set

Fig. 5

True positive decreased slightly, but true negative improved a lot. Though the overall accuracy is slightly lower, we chose to utilize balanced data to train our model because we would like to detect fake reviews as much as possible.



Statistics for Support Vector Machine:				
	Precision	**Recall**	**F1-score**	**Support**
	0.87	0.95	0.91	73499
	0.97	0.92	0.94	129100
avg/total	0.93	0.93	0.93	202599

Fig. 6

Here we have a ROC curve that demonstrates our results.

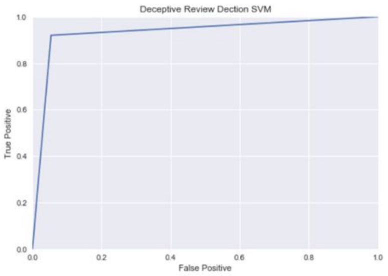


Fig. 7

We also compared the text length distribution between the true review and the fake review we detected.

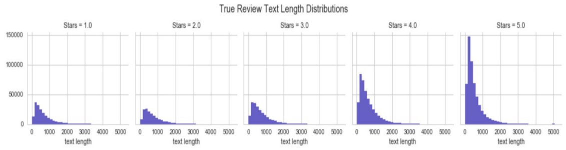


Fig. 8

The pie chart demonstrates fake reviews tend to give more five-star ratings than true reviews would give. As illustrated in the left pie chart, over fifty percent of the fake reviews are five-star reviews.

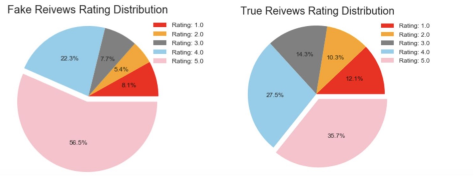


Fig. 9. Pie Chart

b) Information retrieval based on true reviews: Fake reviews created a bias in users' judgments about the quality of a restaurant on Yelp. Thus, filtering out fake reviews is an important step for us to let customers understand a restaurant and let Yelp better serves its original purpose. With the true reviews for a restaurant, we first created a Word Cloud to give customers an overview of a word's appearance frequency in the reviews for a particular restaurant at one glance. As the size of the text grows larger, the word is mentioned more frequently. For instance, below is the word cloud for Chipotle, and words like Mexican, taco, and burrito recur quite often.

After we generated top keywords that affect review sentiment using a support vector machine, we developed functions that could repeat the process for any given merchant and generated a visualization of the result. Here we would like to illustrate words that are more likely to appear in good reviews and bad reviews, which provide strategic value to business owners that enable them to understand customers' demands and preferences.

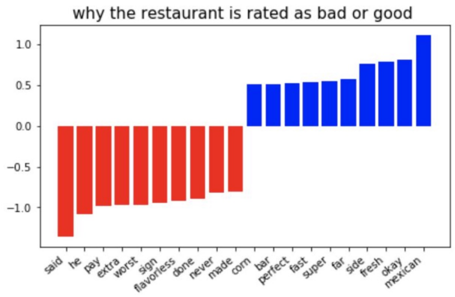


Fig. 10

The bar chart is created using weights obtained from *svm.coef_* and important features based on the absolute size of coefficients. Blue bars correspond to important words in good reviews, and red bars correspond to important words in bad reviews. For instance, using one gram, we can see the words "fresh" and "fast" associated with good reviews.

In addition to the top keywords, we presented the average review score over time in a monthly scope for one particular restaurant.

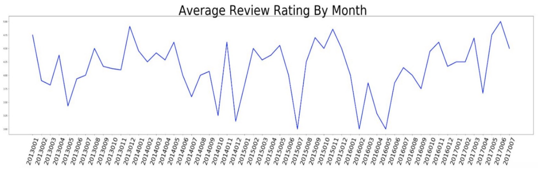
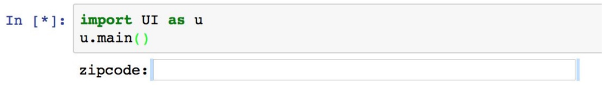


Fig. 11

The graph can present merchants with a timeline of their business performance in any given month of the year. Combined with the top review keywords, merchants could gauge the areas they are doing well and improve their weaknesses. This also enables merchants to notice shifts in customers' attitudes in response to changes in the business. Suppose they have a modification on the menu or service, they can monitor the changes in reviews to reflect its impact in business.

c) Interactive Application: The user interface takes in a restaurant's name and its postal code as the inputs and displays a fake review ratio and a graph demonstrating the top 10 keywords that most positively or negatively influence the reviews of the restaurant. Below is an example that illustrates the process and results:

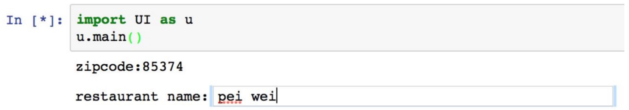


```
In [*]: import UI as u
        u.main()

        zipcode:
```

,

Fig. 12

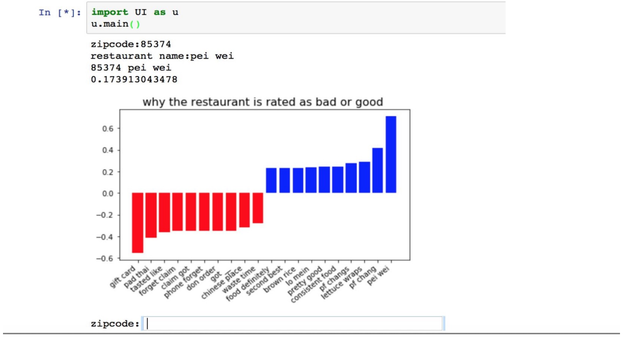


```
In [*]: import UI as u
        u.main()

        zipcode:85374

        restaurant name: pei wei
```

Fig. 13



```
In [*]: import UI as u
        u.main()

        zipcode:85374
        restaurant name:pei wei
        85374 pei wei
        0.173913043478
```

```
zipcode:
```

Fig. 14

VI. FUTURE WORK

1. User Interface - instead of using zip codes, we will improve our user experience by making it more user-friendly by utilizing cities and states inputs with the restaurant name.

2. CNN machine learning model - we currently have unbalanced data for fake reviews, and if we balance the data, CNN will perform worse. For the next step, we will try to supply the CNN model with more logical data. In addition, because of the hardware limitations, we just added one Convolutional Neural Network layer and one Long Short-Term Memory layer. If we can try to add more neural network layers, our model could be more powerful. Moreover, deep Learning can be more efficient with a large dataset. Our dataset is relatively small to fully evoke the power of the deep learning models. We are still exploring methods that can generate more training sets, such as semi-supervised learning, which is known for combining supervised learning

and unsupervised learning models to produce more labeled text data.

3. Fake Review Detection - Intuitively, there will be certain patterns of fake reviews. For example, fake reviews may contain more positive words or be long. If we spend more time on data exploration, we might be able to find such kinds of patterns. If we extract these patterns and give them more weights, the model will be more efficient for training.

4. Sentiment analysis - we only worked with English, but the reviews include Japanese, Chinese, and other languages. It would be great if we could find ways to process these reviews as well. Potential ideas are translating into English or using distinct grammar rules to process the data. Moreover, we would like to improve our model by exploring more efficient feature engineering. In our current model, we only tried bigrams to feed it because of computational limitations. By feeding larger grams into our model, the results may be more accurate.

5. Efficiency of the model needs to be optimized to handle even larger yelp review data in the future. For example, MapReduce implementation on Spark should reduce our training time dramatically. Beyond that, the GPU processor should help improve our model efficiency significantly. After transforming to text data, the matrix representation of our data is pretty similar to the image data. GPU is known for efficiently processing such matrix data. Another great option is Amazon Web Service (AWS), which provides cloud machines with high processing power, many RAMs, and modern GPUs. In the future steps, we could apply this service to our project, which may lead to a more effective result.

6. Database - Exploring other database options, such as MongoDB is possible. MongoDB is also a popular open-source database using dynamic schemas, which implies that we do not have to define the structure of the data before we store them in the database. One of the advantages of utilizing MongoDB is that we can change the records by simply deleting one column or adding one without re-defining a new schema. Since we currently work with large datasets, MongoDB may be easier and more efficient for us to extract, store, and update our data.

VII. CONCLUSION

In this project, our goal is to help consumers and existing and new merchants to use Yelp more efficiently. From consumers' perspective, we enable them to realize the true restaurants' ratings by giving them the ratio of fake reviews. In future works, we can also present the actual restaurants' ratings as one of the outputs. They also can have a better overview of the restaurants on features that the restaurants are famous for or needs to improve on by looking at the keywords, which helps them to choose places that they would most like to dine in. From merchants' point of view, by filtering potential fake reviews, they can find top keywords that indicate the advantages and weaknesses of their restaurants accurately. Combining pricing information and people's preference in different regions, business owners can modify their corporate strategies on pricing, advertising, and dining food and services. In short, our complementary plug-in sample software can help Yelp retain more consumers and merchants by optimizing their searching and matching experience.

VIII. REFERENCE

1.$https : //www.yelp.com/dataset/challenge$
2.$https : //factfinder.census.gov/faces/tableservices/jsf/pages/p$
$ACS\_16\_1YR\_S1902prodType = table$
3.$https : //www.gaslampmedia.com/download - zip -$
$code - latitude - longitude - city - state - county - csv/$
4.$http : //www.ics.uci.edu/ vpsaini/$
5.$http : //www3.cs.stonybrook.edu/ leman/pubs/15 -$
$kdd - collectiveopinionspam.pdf$
6.$https : //plot.ly/python/scatter - plots - on - maps/$